

Attorney Docket No. 1341.1190

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Zhitai SUN et al.

Application No.: Group Art Unit: Unassigned

Filed: February 25, 2004 Examiner: Unassigned

For: METHOD OF AND APPARATUS FOR TASK CONTROL, AND COMPUTER PRODUCT

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-301573

Filed: August 26, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: Feb 25, 2004

By: Mark J. Henry
Mark J. Henry
Registration No. 36,162

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 8月26日
Date of Application:

出願番号 特願2003-301573
Application Number:

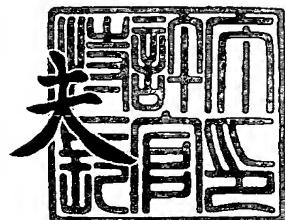
[ST. 10/C] : [JP2003-301573]

出願人 富士通株式会社
Applicant(s):

2003年12月19日

特許庁長官
Commissioner,
Japan Patent Office

今井康



【書類名】 特許願
【整理番号】 0352146
【提出日】 平成15年 8月26日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/46 340
G06F 9/50

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
内
【氏名】 孫 志太

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
内
【氏名】 長谷川 賢一

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
内
【氏名】 加藤 丈治

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
内
【氏名】 内山 徹

【特許出願人】
【識別番号】 000005223
【氏名又は名称】 富士通株式会社

【代理人】
【識別番号】 100089118
【弁理士】
【氏名又は名称】 酒井 宏明

【手数料の表示】
【予納台帳番号】 036711
【納付金額】 21,000円

【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9717671

【書類名】特許請求の範囲**【請求項1】**

オペレーティングシステムを一つのタスクとしてコンピュータに実行させるタスク管理プログラムであって、

前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定する処理有無判定手順と、

前記処理有無判定手順がアイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くする高優先度設定手順と
をコンピュータに実行させることを特徴とするタスク管理プログラム。

【請求項2】

前記高優先度設定手順により優先度が高くされたオペレーティングシステムを所定の時間実行後に、該オペレーティングシステムの優先度を元の優先度に戻す低優先度設定手順をさらにコンピュータに実行させることを特徴とする請求項1に記載のタスク管理プログラム。

【請求項3】

前記処理有無判定手順は、前記オペレーティングシステムの制御下で実行可能状態にあるプロセスがアイドルプロセス以外にあるか否かを判定するプロセス有無判定手順と、

前記オペレーティングシステムに対するスケジュール要求があるか否かを判定するスケジュール要求有無判定手順と、

前記オペレーティングシステムに対する割込み処理要求があるか否かを判定する割込み有無判定手順と
をコンピュータに実行させることを特徴とする請求項1または2に記載のタスク管理プログラム。

【請求項4】

前記高優先度設定手順は、前記処理有無判定手順がアイドルプロセス処理以外に実行する処理があると判定した場合に、所定の時間経過後前記オペレーティングシステムの優先度を高くすることを特徴とする請求項1または2に記載のタスク管理プログラム。

【請求項5】

オペレーティングシステムを一つのタスクとして実行するタスク制御装置であって、

前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定する処理有無判定手段と、

前記処理有無判定手段がアイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くする高優先度設定手段と
を備えたことを特徴とするタスク制御装置。

【書類名】明細書

【発明の名称】タスク管理プログラムおよびタスク制御装置

【技術分野】

【0001】

この発明は、オペレーティングシステムを一つのタスクとしてコンピュータに実行させるタスク管理プログラムおよびタスク制御装置に関し、特に、オペレーティングシステムとの間で負荷処理の均衡化を図り、もってシステム全体のスループットを向上することができるタスク管理プログラムおよびタスク制御装置に関するものである。

【背景技術】

【0002】

近年、1台のコンピュータ上で二つのオペレーティングシステム（以下「OS」という。）を同時に動作させるハイブリッドOSが利用されるようになってきている。これは、OSにはリアルタイム性に優れたものや開発環境に優れたものなどの特徴があり、複数のOSを利用することによって、それぞれのOSの優れた点を使用できる利点があるためである。

【0003】

このようなハイブリッドOSの一つとして、汎用OSをリアルタイムOSの一つのタスクとして実行するOSがある。このハイブリッドOSでは、汎用OSおよびその制御下にあるプロセスはリアルタイム性が低いことから、汎用OSを最低優先度で実行し、他のリアルタイムタスクを優先的に実行することがおこなわれている（たとえば、非特許文献1参照。）。

【0004】

図11は、ハイブリッドOSを説明するための説明図である。同図に示すように、このハイブリッドOSでは、汎用OSはリアルタイムOSの制御プログラム下にあるリアルタイムタスクの一つとして実行される。すなわち、汎用OSは、制御下にあるプロセスも含め汎用OSタスクとして実行される。このような汎用OSとしては、たとえばLinuxがある。

【0005】

【非特許文献1】「リアルタイムOSそれともLinux?」、[平成15年8月8日検索]、インターネット<URL:<http://www.qnx.co.jp/resource/QNX-Linux.pdf>>

【特許文献1】特開平11-149385号

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかしながら、従来のハイブリッドOSでは、汎用OSの制御下にあるプロセスは最低優先度のリアルタイムタスクとして実行されるため、汎用OSの制御下でおこなわれる処理とリアルタイムOSの制御下でおこなわれる処理との間で処理の均衡化が図られていないという問題があった。すなわち、OS間負荷処理均衡化が図られていないという問題があった。

【0007】

そこで、一つのOSでのタスク間処理均衡化手法をハイブリッドOSに用いて処理の均衡化を図ることが考えられる。すなわち、一定の時間内に実行されなかった低優先度の汎用OSタスクの優先度を上げ、汎用OSの制御下で行われる処理を優先することが考えられる。

【0008】

しかし、このようなタスク間処理均衡化手法をハイブリッドOSに単に適用すると、汎用OSはアイドルプロセスを実行しているため常に実行可能な状態にあり、汎用OSタスクの優先度が頻繁に上げられることになる。その結果、過度に汎用OSタスクが優先実行され、他のリアルタイムタスクの動作を妨げるという問題がある。

【0009】

この発明は、上述した従来技術による問題点を解消するためになされたものであり、オペレーティングシステムとの間で負荷処理の均衡化を図り、もってシステム全体のスループットを向上することができるタスク管理プログラムおよびタスク制御装置を提供することを目的とする。

【課題を解決するための手段】

【0010】

上述した課題を解決し、目的を達成するため、本発明は、オペレーティングシステムを一つのタスクとしてコンピュータに実行させるタスク管理プログラムであって、前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定し、アイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くすることを特徴とする。

【0011】

この発明によれば、オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定し、アイドルプロセス処理以外に実行する処理があると判定した場合に、オペレーティングシステムの優先度を高くすることとしたので、オペレーティングシステムとの間で負荷処理の均衡化を図り、もってシステム全体のスループットを向上することができる。また、アイドルプロセス以外の処理がない場合は、優先度を変更せず、無駄な処理が避けられる。

【発明の効果】

【0012】

本発明によれば、オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定し、アイドルプロセス処理以外に実行する処理があると判定した場合に、オペレーティングシステムの優先度を高くするよう構成したので、オペレーティングシステムとの間で負荷処理の均衡化を図り、もってシステム全体のスループットを向上することができるという効果を奏する。

【発明を実施するための最良の形態】

【0013】

以下に添付図面を参照して、この発明に係るタスク管理プログラムおよびタスク制御装置の好適な実施例を詳細に説明する。

【実施例】

【0014】

まず、本実施例に係るハイブリッドOSの構成について説明する。図1は、本実施例に係るハイブリッドOSの構成を示す機能ブロック図である。同図に示すように、このハイブリッドOSは、割込みハンドラ100と、リアルタイムOS200と、汎用OS300とを有する。なお、同図において、実線の矢印は制御の流れを示し、破線の矢印はデータの参照を示す。

【0015】

割込みハンドラ100は、ハードウェア10からの割込みを処理する処理部であり、リアルタイムOS処理向け割込みハンドラ110と、汎用OS処理向け割込みハンドラ120と、タイマ割込みハンドラ130とを有する。なお、割込みハンドラ100は、リアルタイムOS200の一部であるが、ここでは、リアルタイムOS200の他の部分と区別して割込みハンドラ100として示している。

【0016】

リアルタイムOS処理向け割込みハンドラ110は、ハードウェア10からの割込みを処理する処理部であり、割込みがリアルタイムOS200に対する割込みであるか汎用OS300に対する割込みであるかを判定し、リアルタイムOS200に対する割込みである場合にはその割込みの処理をおこない、汎用OS300に対する割込みである場合には、汎用OS処理向け割込みハンドラ120に制御を移す。

【0017】

汎用OS処理向け割込みハンドラ120は、汎用OS300に対する割込みを処理する

処理部である。この汎用OS処理向け割込みハンドラ120は、緊急処理だけをおこなつて汎用OS300の割込み処理要求フラグ320をセットし、汎用OS300に対して割込み処理を要求する。すなわち、汎用OS300は、割込み処理要求フラグ320が設定されると、割込み処理をおこなう。

【0018】

タイマ割込みハンドラ130は、タイマ割込みを処理する処理部である。リアルタイムOS処理向け割込みハンドラ110は、割込みがタイマ割込みである場合に、このタイマ割込みハンドラ130に制御を移す。

【0019】

このタイマ割込みハンドラ130は、通常のディスパッチ処理に加えて、汎用OS300の処理が必要であるか否かを判定し、汎用OS300の処理が必要であると判定した場合には、汎用OS300の優先度を高い優先度に変更する。

【0020】

このタイマ割込みハンドラ130が汎用OS300の優先度を変更することによって、優先度の低い汎用OS300の負荷処理時間を確保し、リアルタイムOS200と汎用OS300との間で負荷処理の均衡化を図ることができる。なお、このタイマ割込みハンドラ130の処理の詳細については後述する。

【0021】

リアルタイムOS200は、汎用OS300をリアルタイムタスクの一つとして実行するOSであり、システムパラメータ記憶部210と、TCBリスト220と、スケジューラ230と、システムコール処理部240とを有する。

【0022】

システムパラメータ記憶部210は、リアルタイムOS200の実行に必要なシステムパラメータを記憶した記憶部である。図2は、OS間負荷処理均衡化のためにリアルタイムOS200に追加したシステムパラメータを示す図である。

【0023】

同図に示すように、OS間負荷処理均衡化のためにリアルタイムOS200に追加したシステムパラメータとしては、均衡化優先度401と、均衡化実行時間402と、持続タイムアウト値403とがある。

【0024】

均衡化優先度401は、OS間負荷処理均衡化のために汎用OS300に設定される優先度であり、通常の優先度より高い値である。汎用OS300の優先度をこの均衡化優先度401に上げることによって、汎用OS300は、リアルタイムタスクとしての優先度が高くなり、実行時間を確保することができる。

【0025】

均衡化実行時間402は、汎用OS300が均衡化優先度401で実行される時間である。持続タイムアウト値403は、汎用OS300の優先度を均衡化優先度401に変更する必要が生じた際に均衡化優先度401に変更するまでの時間である。汎用OS300は、基本的には低い優先度で実行されることから、優先度を均衡化優先度401に変更する必要が生じた場合でも、すぐに実行される必要はなく、この持続タイムアウト値403の経過後に始めて優先度が高く設定される。

【0026】

TCBリスト220は、各リアルタイムタスクの制御情報を記憶したTCB(Task Control Block)のリストである。図3は、OS間負荷処理均衡化のためにTCBに設けたパラメータを示す図である。同図に示すように、このTCBは、タスク生成時の優先度を示す本優先度501と、タスクが実行されるときの実優先度502と、汎用OS300が均衡化優先度401で実行される時間を測定する均衡化実行タイマ503と、持続タイムアウト値403を測定する持続タイマ504とを有する。

【0027】

スケジューラ230は、リアルタイムタスクの実行を管理する処理部であり、OS間負

荷処理均衡化によって実行されている汎用OS300に対しては、資源待ちになった場合に実優先度502を元の本優先度501に戻す処理などをおこなう。

【0028】

システムコール処理部240は、リアルタイムタスクからのシステムコールを処理する処理部であり、システムコールの一つとしてリアルタイムタスクによるOS間負荷処理均衡化要求の処理をおこなう。

【0029】

汎用OS300は、複数のプロセスを並行処理するOSであるとともに、リアルタイムOS200の制御下で実行されるリアルタイムタスクである。この汎用OS300は、配下にアイドルプロセスを有し、このアイドルプロセスは常に実行可能な状態にある。なお、この汎用OS300は、配下のプロセスとともにリアルタイムOS200の制御下で汎用OSタスク400として実行される。

【0030】

この汎用OS300は、PCBリスト310と、割込み処理要求フラグ320と、汎用OSスケジューラ330とを有する。PCBリスト310は、各プロセスの制御情報を記憶したPCB(Process Control Block)のリストである。

【0031】

割込み処理要求フラグ320は、汎用OS300による割込み処理が必要な場合に、汎用OS処理向け割込みハンドラ120によって設定されるフラグである。なお、この割込み処理要求フラグ320は、汎用OS300のグローバル領域に設けられ、タイマ割込みハンドラ130が参照することができる。

【0032】

汎用OSスケジューラ330は、PCBリスト310や割込み処理要求フラグ320の情報を参照して汎用OS300の制御下で実行されるプロセスの実行を管理する処理部である。

【0033】

次に、本実施例に係るOS間負荷処理均衡化の処理について説明する。まず、図1に示したタイマ割込みハンドラ130に追加したOS間負荷処理均衡化の処理について説明する。図4は、図1に示したタイマ割込みハンドラ130に追加したOS間負荷処理均衡化の処理手順を示すフローチャートである。

【0034】

同図に示すように、タイマ割込みハンドラ130に追加したOS間負荷処理均衡化の処理では、汎用OSタスク400が実行中であるか否かを判定し（ステップS401）、実行中でない場合には、汎用OSタスク400の処理が必要か否かを判定する（ステップS402）。

【0035】

そして、汎用OSタスク400の処理が必要である場合には、持続タイマ504の値が正であるか否かを判定し（ステップS403）、持続タイマ504の値が正である場合には、持続タイマ504が動作している場合であるので、持続タイマ504の値を「1」減らし（ステップS404）、減らした結果、持続タイマ504の値が「0」になったか否かを判定する（ステップS405）。

【0036】

そして、持続タイマ504の値が「0」になった場合には、汎用OSタスク400の優先度を上げて処理する必要があるので、実優先度502に均衡化優先度401を設定し、均衡化実行タイマ503に均衡化実行時間402を設定し（ステップS406）、OS間負荷処理均衡化のためのタイマ割込み処理を終了する。

【0037】

一方、持続タイマ504の値が「0」にならない場合には、持続タイマ504の値が「0」になるまで汎用OSタスク400の優先度を上げる必要はないので、OS間負荷処理均衡化のためのタイマ割込み処理を終了する。

【0038】

また、持続タイマ504の値が正でない場合にも、汎用OSタスク400の優先度を上げる処理は既におこなわれた場合であるので、OS間負荷処理均衡化の処理を終了する。また、汎用OSタスク400の処理が必要でない場合にも、OS間負荷処理均衡化のためのタイマ割込み処理を終了する。

【0039】

また、汎用OSタスク400が実行中である場合には、汎用OSタスク400の実優先度502を調べ、均衡化優先度401で実行されているか否かを判定し（ステップS407）、均衡化優先度401で実行されている場合には、均衡化実行タイマ503の値を「1」減らし（ステップS408）、減らした結果、均衡化実行タイマ503の値が「0」になったか否かを判定する（ステップS409）。

【0040】

そして、均衡化実行タイマ503の値が「0」になった場合には、均衡化優先度401で汎用OSタスク400を実行する時間が終了した場合であるので、実優先度502を元の本優先度501に戻し、持続タイマ504に持続タイムアウト値403を設定する（ステップS410）。

【0041】

一方、均衡化実行タイマ503の値が「0」にならなかった場合には、均衡化優先度401のまま汎用OSタスク400を実行する必要があるため、実優先度502を変更することなくOS間負荷処理均衡化のためのタイマ割込み処理を終了する。また、汎用OSタスク400が均衡化優先度401で実行されていない場合にも、OS間負荷処理均衡化のためのタイマ割込み処理を終了する。

【0042】

このように、タイマ割込みハンドラ130が汎用OSタスク400の処理が必要か否かを判定し、汎用OSタスク400の処理が必要である場合には、持続タイマ504に設定した持続タイムアウト値403の時間が経過した場合に汎用OSタスク400の実優先度502を均衡化優先度401に上げることとしたので、汎用OSタスク400の実行時間を確保することができ、もってOS間負荷処理の均衡化を図ることができる。

【0043】

なお、ここでは、汎用OSタスク400が均衡化優先度401で実行された場合に均衡化実行タイマ503の値を「1」減らすこととしたが、汎用OSタスク400の実優先度502が均衡化優先度401に変更された場合に均衡化実行タイマ503の値を「1」減らすこともできる。

【0044】

また、ここでは、均衡化実行タイマ503の値をタイマ割り込みのたびに「1」ずつ減らしていく、均衡化実行タイマ503の値が「0」になる場合、すなわち、汎用OSタスク400が均衡化実行時間を消費してしまう場合について説明したが、均衡化実行タイマ503の値は、汎用OSタスク400が資源待ちの状態になったとき、あるいは汎用OSタスク400の実行中により優先度の高いリアルタイムタスクに実行権が横取りされた場合にも「0」クリアされる。

【0045】

次に、汎用OSタスク400の処理が必要であるか否かの判定処理について、図5および図6を用いて説明する。汎用OSタスク400の処理が必要である場合としては、アイドルプロセス以外に実行可能プロセスがある場合と、アイドルプロセスだけが実行可能であってもスケジュール要求または割込み処理要求がある場合とがある。

【0046】

したがって、汎用OSタスク400の処理が必要であるか否かを判定するためには、汎用OS300の制御下で実行されるプロセスに関する情報を取得する必要がある。そこで、汎用OS300の制御下で実行されるプロセスに関する情報を取得する手順について説明する。

【0047】

図5は、汎用OS300制御下のプロセスの情報を取得する手順を説明するための説明図である。同図に示すように、各TCBにはカレントスタックポインタが格納されており、このカレントスタックポインタの値からスタックの先頭アドレスを取得することができる。なぜならば、プロセス毎に同サイズで数キロバイト単位のスタックが割り当てられる。例えば、スタックを8キロバイトの大きさとしたとき、スタックの先頭アドレスは8キロバイトの倍数を割り当ててあるため、カレントスタックポインタの値の、8キロバイトより小さいアドレスを示す部分を「0」とすることによって、スタックの先頭アドレスを取得することができる。

【0048】

また、スタックの先頭には、汎用OS300制御下のプロセスのPCBの先頭アドレスが格納されている。そして、PCBの先頭アドレスから所定の値だけ先のアドレスには、プロセスID(PID)601が格納されており、このPID601によってこのプロセスがアイドルプロセスであるか否かを判定することができる。

【0049】

また、PID601の格納アドレスからさらに一定の値だけ先のアドレスには、スケジュール要求フラグ602が格納されている。このスケジュール要求フラグ602は、汎用OS300に対してスケジュール要求がある場合にセットされる。

【0050】

このように、PCBには、PID601およびスケジュール要求フラグ602が格納されており、タイマ割込みハンドラ130は、これらのPID601およびスケジュール要求フラグ602と、図1に示した割込み処理要求フラグ320とを用いて汎用OSタスク400の実行が必要であるか否かを判定することができる。

【0051】

そこで、汎用OSタスク400の処理が必要であるか否かを判定する処理手順について説明する。図6は、汎用OSタスク処理必要性判定処理の処理手順を示すフローチャートである。同図に示すように、この汎用OSタスク処理必要性判定処理は、まずTCBからカレントスタックポインタを取得する(ステップS601)。

【0052】

そして、カレントスタックポインタからスタックの先頭アドレスを求め、スタックの先頭に格納されたPCB先頭アドレスを取得する(ステップS602)。このPCB先頭アドレスは、最後に実行されたプロセスのPCB先頭アドレスである。

【0053】

そして、PCB先頭アドレスに一定の値を加えてPID601の格納アドレスを算出し(ステップS603)、算出したアドレスからPID601を読み出してこのプロセスがアイドルプロセスであるか否かを判定する(ステップS604)。

【0054】

その結果、アイドルプロセスである場合には、スケジュール要求フラグ602の格納アドレスを算出し(ステップS605)、算出したアドレスからスケジュール要求フラグ602を読み出してスケジュール要求があるか否かを判定する(ステップS606)。

【0055】

そして、スケジュール要求がない場合には、割込み処理要求フラグ320を読み出し(ステップS607)、割込み要求があるか否かを判定する(ステップS608)。その結果、割込み要求がない場合には、汎用OSタスク処理不要と判定し(ステップS609)、割込み要求がある場合には、汎用OSタスク処理必要と判定する(ステップS610)。また、スケジュール要求がある場合またはアイドルプロセスでない場合には、汎用OSタスク処理必要と判定する(ステップS610)。

【0056】

このように、タイマ割込みハンドラ130がPCBに格納されたPID601およびスケジュール要求フラグ602ならびに汎用OS300のグローバル領域に格納された割込

み処理要求フラグ320を用いて汎用OSタスク400の処理が必要か否かを判定することによって、汎用OSタスク400の処理が必要な場合にだけ汎用OSタスク400の優先度を上げることができ、不要な汎用OSタスク400の処理を防ぐことができる。

【0057】

次に、OS間負荷処理均衡化により実行中の汎用OSタスク400が資源待ちになったときのスケジューラ230の処理について説明する。図7は、OS間負荷処理均衡化により実行中の汎用OSタスク400が資源待ちになったときのスケジューラ230の処理手順を示すフローチャートである。

【0058】

同図に示すように、OS間負荷処理均衡化により実行中の汎用OSタスク400が資源待ちになると、スケジューラ230は、OS間負荷処理均衡化を中止する処理をおこなう。すなわち、スケジューラ230は、実優先度502を本優先度501に戻し（ステップS701）、均衡化実行タイマ503を「0」クリアする（ステップS702）。そして、汎用OSタスク400の状態を実行状態から処理必要状態に変更する（ステップS703）。

【0059】

次に、OS間負荷処理均衡化により実行中の汎用OSタスク400の実行権が他のリアルタイムタスクによって横取りされるときのスケジューラ230の処理について説明する。図8は、OS間負荷処理均衡化により実行中の汎用OSタスク400の実行権が他のリアルタイムタスクによって横取りされるときのスケジューラ230の処理手順を示すフローチャートである。

【0060】

同図に示すように、OS間負荷処理均衡化により実行中の汎用OSタスク400の実行権が他のリアルタイムタスクによって横取りされると、スケジューラ230は、残された均衡化実行時間402の均衡化処理を打切るか否かを判定する（ステップS801）。ここで、残された均衡化実行時間402の均衡化処理を打切るか否かは、スケジューリング方式として変更することが可能である。

【0061】

そして、残された均衡化実行時間402の均衡化処理を打切る場合には、実優先度502を本優先度501に戻し（ステップS802）、持続タイマ504に持続タイムアウト値403を設定し、持続タイマ504を起動する（ステップS803）。そして、均衡化実行タイマ503を「0」クリアし（ステップS804）、汎用OSタスク400の状態を実行状態から処理必要状態に変更する（ステップS805）。

【0062】

一方、残された均衡化実行時間402の均衡化処理を打切らない場合には、実優先度502、持続タイマ504および均衡化実行タイマ503を変更することなく、汎用OSタスク400の状態を実行状態から処理必要状態に変更する（ステップS805）。

【0063】

次に、図1に示したシステムコール処理部240によるOS間負荷処理均衡化要求に対する処理について説明する。図9は、図1に示したシステムコール処理部240によるOS間負荷処理均衡化要求に対する処理手順を示すフローチャートである。

【0064】

同図に示すように、システムコール処理部240は、OS間負荷処理均衡化要求があると、汎用OSタスク400を処理する必要があるか否かを判定する（ステップS901）。なお、この判定は、図6に示した処理手順によっておこなわれる。

【0065】

そして、汎用OSタスク400を処理する必要がある場合には、実優先度502を均衡化優先度401に上げ（ステップS902）、均衡化実行タイマ503に均衡化実行時間402を設定する（ステップS903）。

【0066】

そして、持続タイマ504を「0」クリアし（ステップS904）、ディスパッチをおこなう（ステップS905）。ここで、汎用OSタスク400の優先度が最も高くなった場合には、ディスパッチによって汎用OSタスク400が実行されることになる。一方、汎用OSタスク400を処理する必要がない場合には、そのまま処理を終了する。

【0067】

次に、OS間負荷処理均衡化に関する汎用OSタスク400の状態遷移について説明する。図10は、OS間負荷処理均衡化に関する汎用OSタスク400の状態遷移を示す図である。同図に示すように、汎用OSタスク400の状態としては、初期状態（S1）と、優先度が本優先度501である処理必要状態（S2）と、優先度が均衡化優先度401である処理必要状態（S3）と、実行状態（S4）と、アイドル状態（S5）とがある。

【0068】

また、汎用OSタスク400の状態を遷移させるイベントまたは条件としては、タスク生成（E1）と、タイマ割込み（E2）と、持続タイマ504に設定された持続タイムアウト値の時間経過（E3）と、タスクスイッチング（E4）と、均衡化実行時間経過（E5）と、資源待ち（E6）と、資源待ち解除（E7）と、均衡化要求システムコール（E8）とがある。

【0069】

汎用OSタスク400は、最初は初期状態（S1）にあり、タスクとして生成されると優先度が本優先度501である処理必要状態（S2）に遷移する。また、持続タイマ504に持続タイムアウト値403を設定し、持続タイマ504を起動する。そして、このS2の状態でタイマ割込み（E2）が発生するたびに、リアルタイムOSは、持続タイマ504の値を「1」ずつ減らしていく。なお、ここでのリアルタイムOSは、図1に示した割込みハンドラ100とリアルタイムOS200とを含むものとする。

【0070】

そして、持続タイマ504の値が「0」になると、リアルタイムOSは、実優先度502を均衡化優先度401に変更し、均衡化実行タイマ503に均衡化実行時間402を設定する。また、汎用OSタスク400は、優先度が均衡化優先度401である処理必要状態（S3）に遷移する。

【0071】

また、汎用OSタスク400がS2の状態にある場合に、タスクスイッチング（E4）によって汎用OSタスク400に実行権が与えられると、汎用OSタスク400は実行状態（S4）に遷移する。この時、リアルタイムOSは、持続タイマ504を停止状態とする。

【0072】

また、均衡化要求システムコール（E8）があると、リアルタイムOSは、実優先度502を均衡化優先度401に変更し、均衡化実行タイマ503に均衡化実行時間402を設定する。そして、汎用OSタスク400は、優先度が均衡化優先度401である処理必要状態（S3）に遷移する。

【0073】

また、汎用OSタスク400が、優先度が均衡化優先度401である処理必要状態（S3）である場合にタスクスイッチング（E4）が発生し、汎用OSタスク400に実行権が与えられると、汎用OSタスク400は実行状態（S4）に遷移する。この時、リアルタイムOSは、均衡化実行タイマ503の作動を開始する。

【0074】

また、汎用OSタスク400が実行状態（S4）にある場合には、均衡化実行中であれば、タイマ割込み（E2）が発生するたびに、リアルタイムOSは、均衡化実行タイマ503の値を「1」ずつ減らしていく。

【0075】

そして、均衡化実行タイマ503の値が「0」になると、リアルタイムOSは、実優先度502を本優先度501に戻し、ディスパッチをおこなう。また、汎用OSタスク40

0は、優先度が本優先度501である処理必要状態（S2）に遷移する。

【0076】

また、汎用OSタスク400が実行状態（S4）でタスクスイッチング（E4）が発生した場合には、リアルタイムOSは、実優先度502を本優先度501に戻し、持続タイマ504に持続タイムアウト値403を設定し、持続タイマ504を起動する。また、汎用OSタスク400は、優先度が本優先度501である処理必要状態（S2）に遷移する。

【0077】

また、汎用OSタスク400が実行状態（S4）で資源待ち（E6）が発生した場合には、リアルタイムOSは、実優先度502を本優先度501に戻し、均衡化実行タイマ503を停止する。また、汎用OSタスク400は、アイドル状態（S5）に遷移する。

【0078】

そして、アイドル状態（S5）で資源待ち解除（E7）が発生すると、リアルタイムOSは、持続タイマ504に持続タイムアウト値403を設定し、持続タイマ504を起動する。また、汎用OSタスク400は、優先度が本優先度501である処理必要状態（S2）に遷移する。

【0079】

このように、汎用OSタスク400の各状態において発生するイベントに対応してリアルタイムOSが汎用OSタスク400の実優先度502を変更するとともに、持続タイマ504および均衡化実行タイマ503を操作することによって、OS間負荷処理の均衡化を図ることができる。

【0080】

上述してきたように、本実施例では、タイマ割込みハンドラ130がP.CBリスト310および割込み処理要求フラグ320を用いて汎用OSタスク400の処理の必要性を判定し、汎用OSタスク400の処理が必要な場合にだけ、汎用OSタスク400の優先度を上げることとしたので、不要に汎用OSタスク400に実行権が与えられることを防ぎ、OS間負荷処理の均衡化を図ることができる。

【0081】

（付記1）オペレーティングシステムを一つのタスクとしてコンピュータに実行させるタスク管理プログラムであって、

前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定する処理有無判定手順と、

前記処理有無判定手順がアイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くする高優先度設定手順と

をコンピュータに実行させることを特徴とするタスク管理プログラム。

【0082】

（付記2）前記処理有無判定手順および前記高優先度設定手順を実行するシステムコールを備えたことを特徴とする付記1に記載のタスク管理プログラム。

【0083】

（付記3）前記高優先度設定手順により優先度が高くされたオペレーティングシステムを所定の時間実行後に、該オペレーティングシステムの優先度を元の優先度に戻す低優先度設定手順をさらにコンピュータに実行させることを特徴とする付記1または2に記載のタスク管理プログラム。

【0084】

（付記4）前記処理有無判定手順は、前記オペレーティングシステムの制御下で実行可能状態にあるプロセスがアイドルプロセス以外にあるか否かを判定するプロセス有無判定手順と、

前記オペレーティングシステムに対するスケジュール要求があるか否かを判定するスケジュール要求有無判定手順と、

前記オペレーティングシステムに対する割込み処理要求があるか否かを判定する割込み

有無判定手順と

をコンピュータに実行させることを特徴とする付記1または2に記載のタスク管理プログラム。

【0085】

(付記5) 前記プロセス有無判定手順は、前記オペレーティングシステムの制御下で実行されるプロセスのプロセス制御ブロックに格納されたプロセス識別子を用いて、実行可能状態にあるプロセスがアイドルプロセス以外にあるか否かを判定することを特徴とする付記4に記載のタスク管理プログラム。

【0086】

(付記6) 前記スケジュール要求有無判定手順は、前記オペレーティングシステムの制御下で実行されるプロセスのプロセス制御ブロックに格納されたスケジュール要求フラグを用いて、スケジュール要求があるか否かを判定することを特徴とする付記4に記載のタスク管理プログラム。

【0087】

(付記7) 前記割込み有無判定手順は、前記オペレーティングシステムのグローバル領域に設けられた割込み処理要求フラグを用いて、割込み処理要求があるか否かを判定することを特徴とする付記4に記載のタスク管理プログラム。

【0088】

(付記8) 前記高優先度設定手順は、前記処理有無判定手順がアイドルプロセス処理以外に実行する処理があると判定した場合に、所定の時間経過後前記オペレーティングシステムの優先度を高くすることを特徴とする付記1または2に記載のタスク管理プログラム。

【0089】

(付記9) オペレーティングシステムを一つのタスクとして実行するタスク制御装置であって、

前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定する処理有無判定手段と、

前記処理有無判定手段がアイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くする高優先度設定手段と

を備えたことを特徴とするタスク制御装置。

【0090】

(付記10) オペレーティングシステムを一つのタスクとして実行するタスク制御方法であって、

前記オペレーティングシステムの制御下で実行される処理の中にアイドルプロセス処理以外に実行する処理があるか否かを判定する処理有無判定工程と、

前記処理有無判定工程がアイドルプロセス処理以外に実行する処理があると判定した場合に、前記オペレーティングシステムの優先度を高くする高優先度設定工程と

を含んだことを特徴とするタスク制御方法。

【産業上の利用可能性】

【0091】

以上のように、本発明にかかるタスク管理プログラムおよびタスク制御装置は、ハイブリッドOSを用いるコンピュータシステムに有用であり、特に、汎用OSをリアルタイムタスクの一つとして動作させるハイブリッドOSを使用するコンピュータシステムや組み込み機器などに適している。

【図面の簡単な説明】

【0092】

【図1】本実施例に係るハイブリッドOSの構成を示す機能ブロック図である。

【図2】OS間負荷処理均衡化のためにリアルタイムOSに追加したシステムパラメータを示す図である。

【図3】OS間負荷処理均衡化のためにTCBに設けたパラメータを示す図である。

【図4】図1に示したタイマ割込みハンドラに追加したOS間負荷処理均衡化の処理

手順を示すフローチャートである。

【図5】汎用OS制御下のプロセスの情報を取得する手順を説明するための説明図である。

【図6】汎用OSタスク処理必要性判定処理の処理手順を示すフローチャートである。

【図7】OS間負荷処理均衡化により実行中の汎用OSタスクが資源待ちになったときのスケジューラの処理手順を示すフローチャートである。

【図8】OS間負荷処理均衡化により実行中の汎用OSタスクの実行権が他のリアルタイムタスクによって横取りされるときのスケジューラの処理手順を示すフローチャートである。

【図9】図1に示したシステムコール処理部によるOS間負荷処理均衡化要求に対する処理手順を示すフローチャートである。

【図10】OS間負荷処理均衡化に関する汎用OSタスクの状態遷移を示す図である。

【図11】ハイブリッドOSを説明するための説明図である。

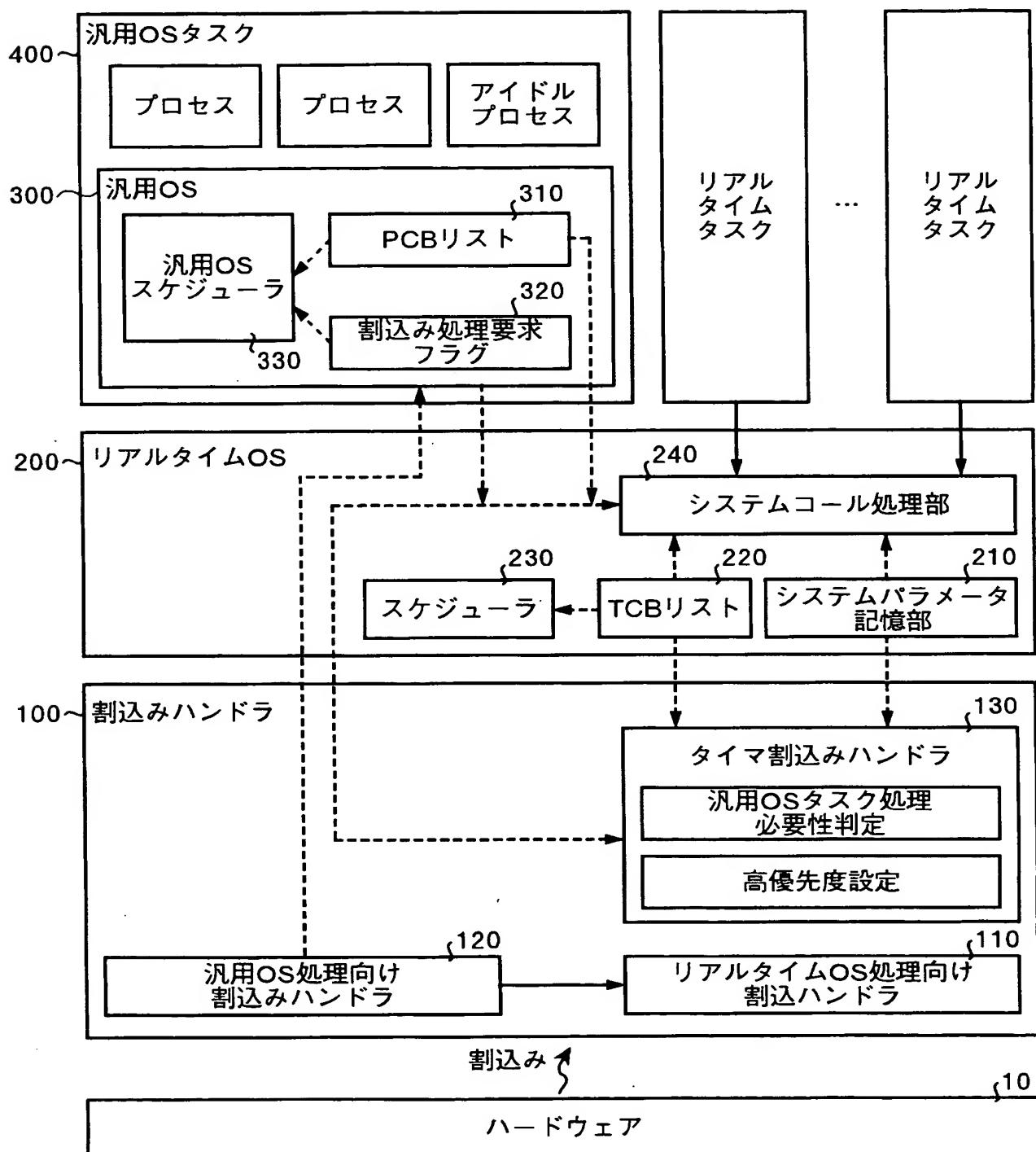
【符号の説明】

【0093】

1 0	ハードウェア
1 0 0	割込みハンドラ
1 1 0	リアルタイムOS処理向け割込みハンドラ
1 2 0	汎用OS処理向け割込みハンドラ
1 3 0	タイマ割込みハンドラ
2 0 0	リアルタイムOS
2 1 0	システムパラメータ記憶部
2 2 0	TCBリスト
2 3 0	スケジューラ
2 4 0	システムコール処理部
3 0 0	汎用OS
3 1 0	PCBリスト
3 2 0	割込み処理要求フラグ
3 3 0	汎用OSスケジューラ
4 0 0	汎用OSタスク
4 0 1	均衡化優先度
4 0 2	均衡化実行時間
4 0 3	持続タイムアウト値
5 0 1	本優先度
5 0 2	実優先度
5 0 3	均衡化実行タイマ
5 0 4	持続タイマ
6 0 1	PID
6 0 2	スケジュール要求フラグ

【書類名】 図面
【図 1】

本実施例に係るハイブリットOSの構成を示す機能ブロック図



【図 2】

OS間負荷処理均衡化のために
リアルタイムOSに追加したシステムパラメータ

均衡化優先度	～401
均衡化実行時間	～402
持続タイムアウト値	～403

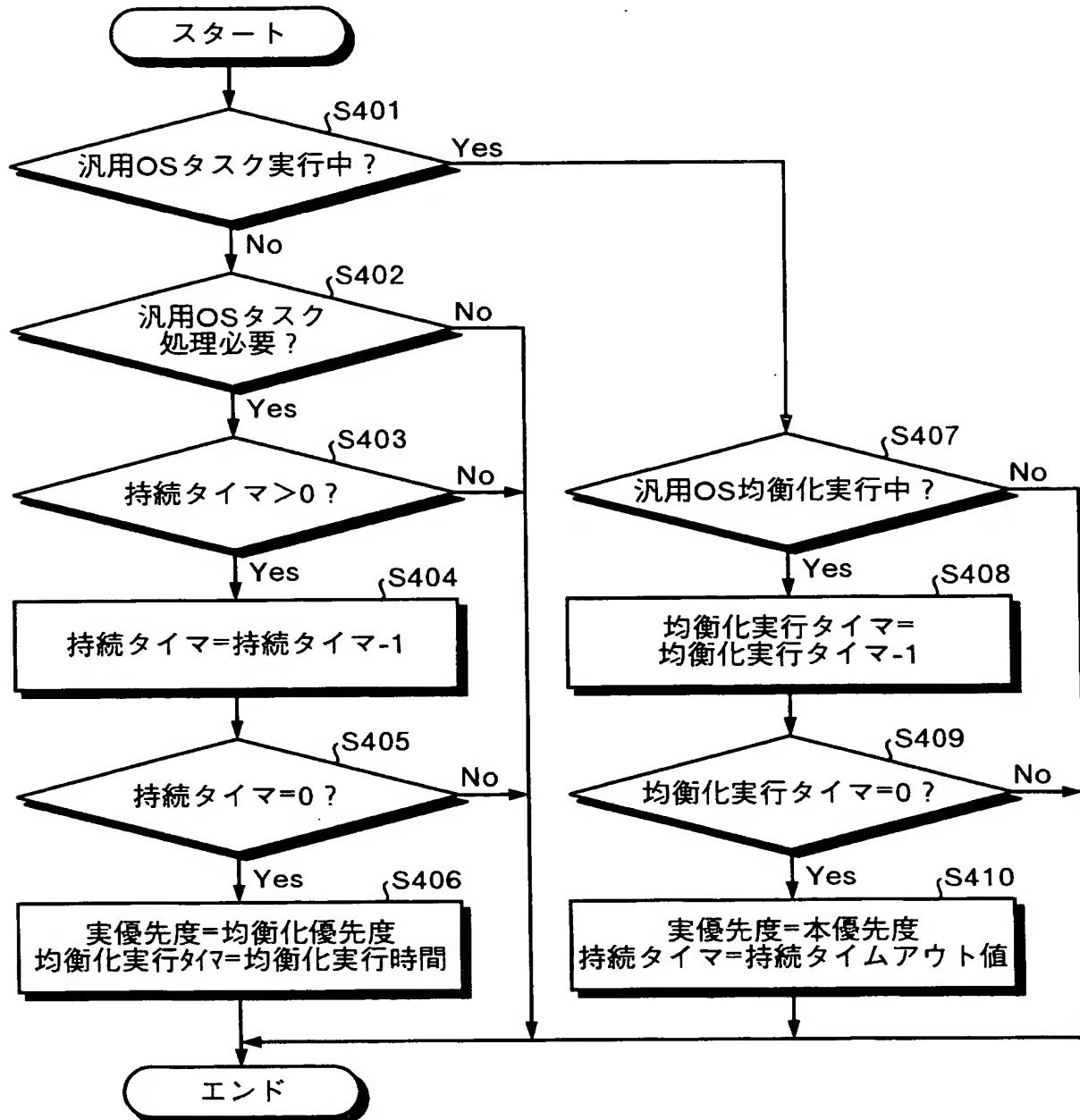
【図 3】

OS間負荷処理均衡化のためにTCBに設定したパラメータ

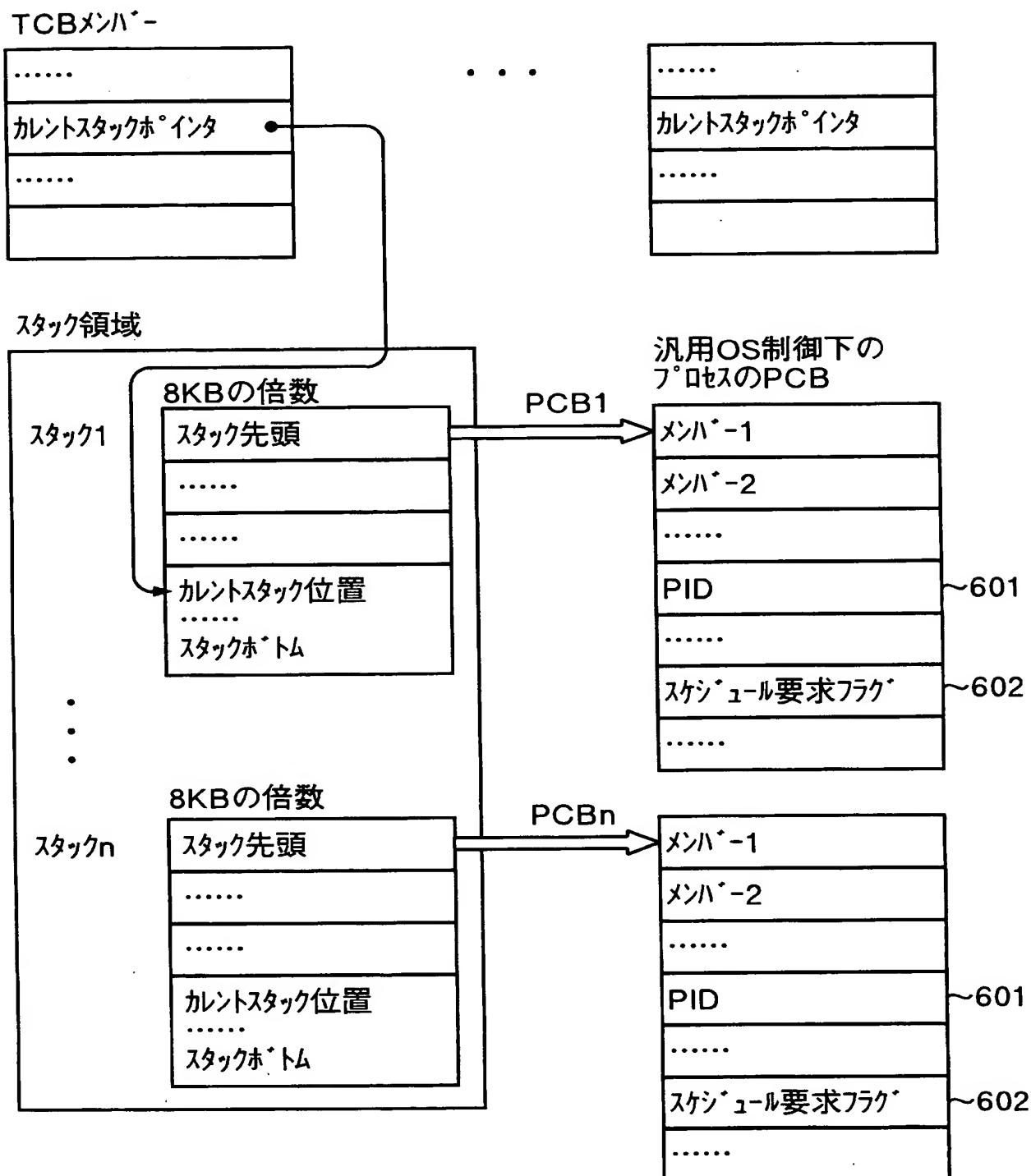
本優先度	～501
実優先度	～502
均衡化実行タイマ	～503
持続タイマ	～504

【図4】

図1に示したタイマ割込みハンドラに追加した
OS間負荷処理均衡化の処理手順を示すフローチャート

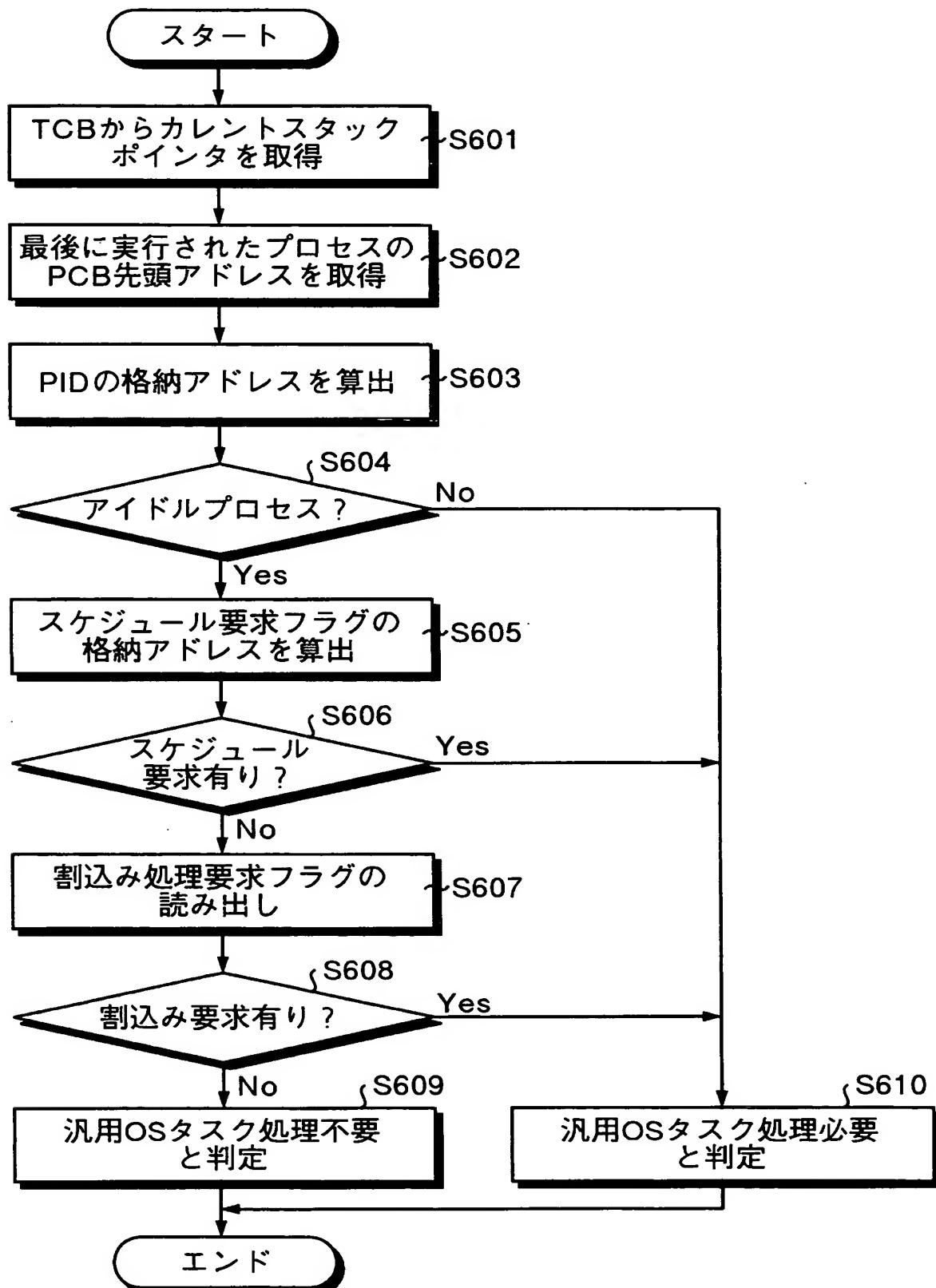


【図 5】

汎用OS制御下のプロセスの情報を
取得する手順を説明するための説明図

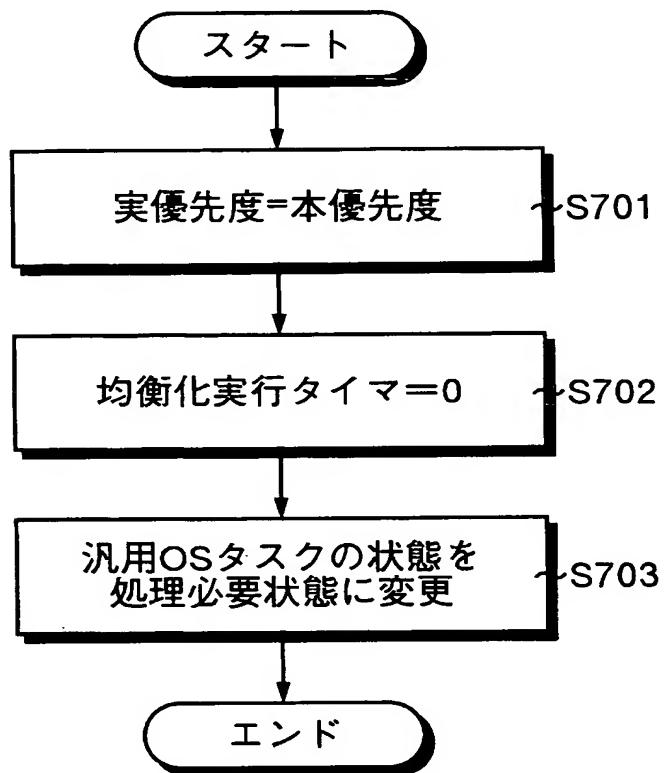
【図6】

汎用OSタスク実行必要性判定処理の処理手順を示すフローチャート



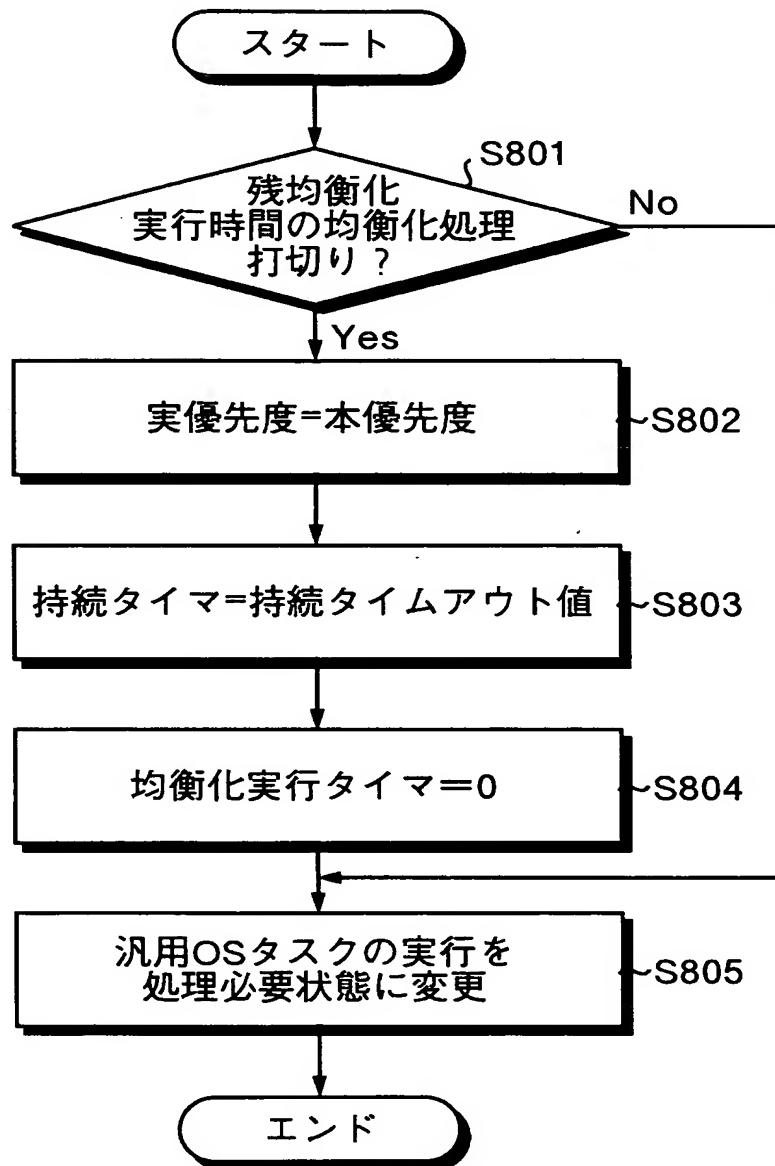
【図7】

OS間負荷処理均衡化により実行中の汎用OSタスクが
資源待ちになったときのスケジューラの処理手順を示すフローチャート



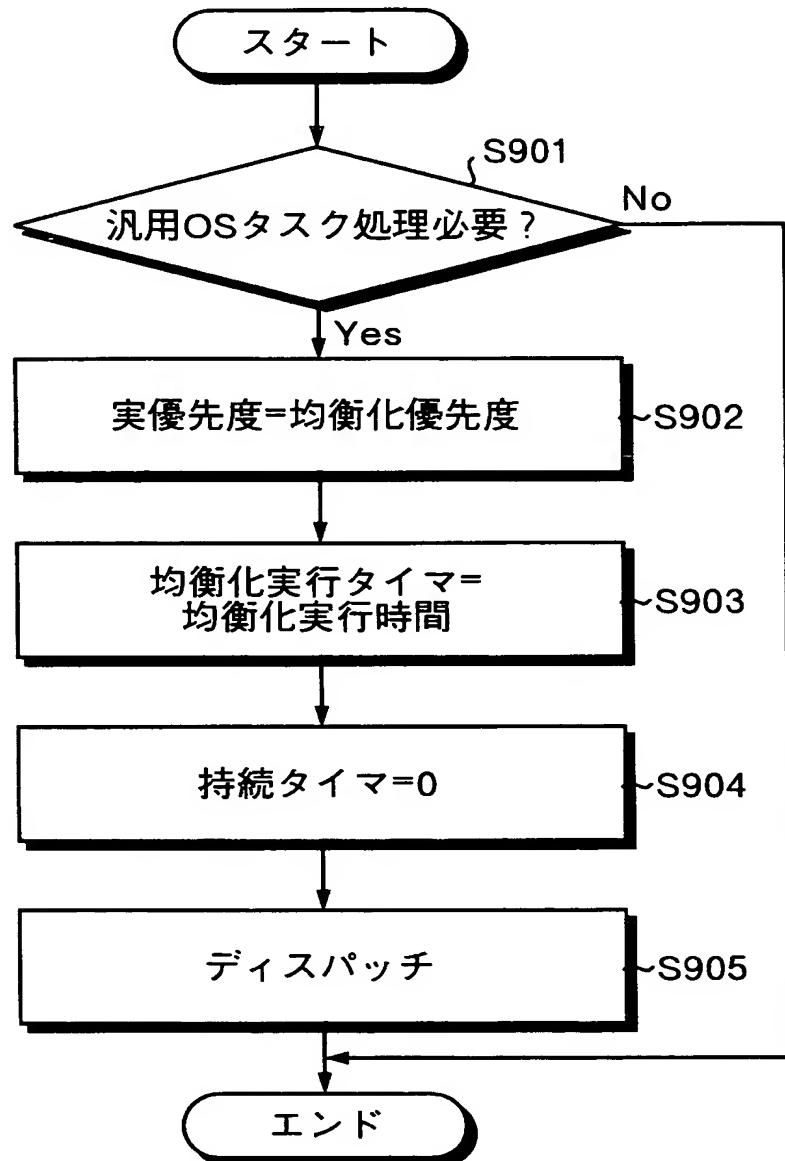
【図 8】

OS間負荷処理均衡化により実行中の
汎用OSタスクの実行権が他のリアルタイムタスクに
横取りされたときのスケジューラの処理手順を示すフローチャート



【図9】

図1に示したシステムコール処理部による
OS間負荷処理均衡化の処理手順を示すフローチャート



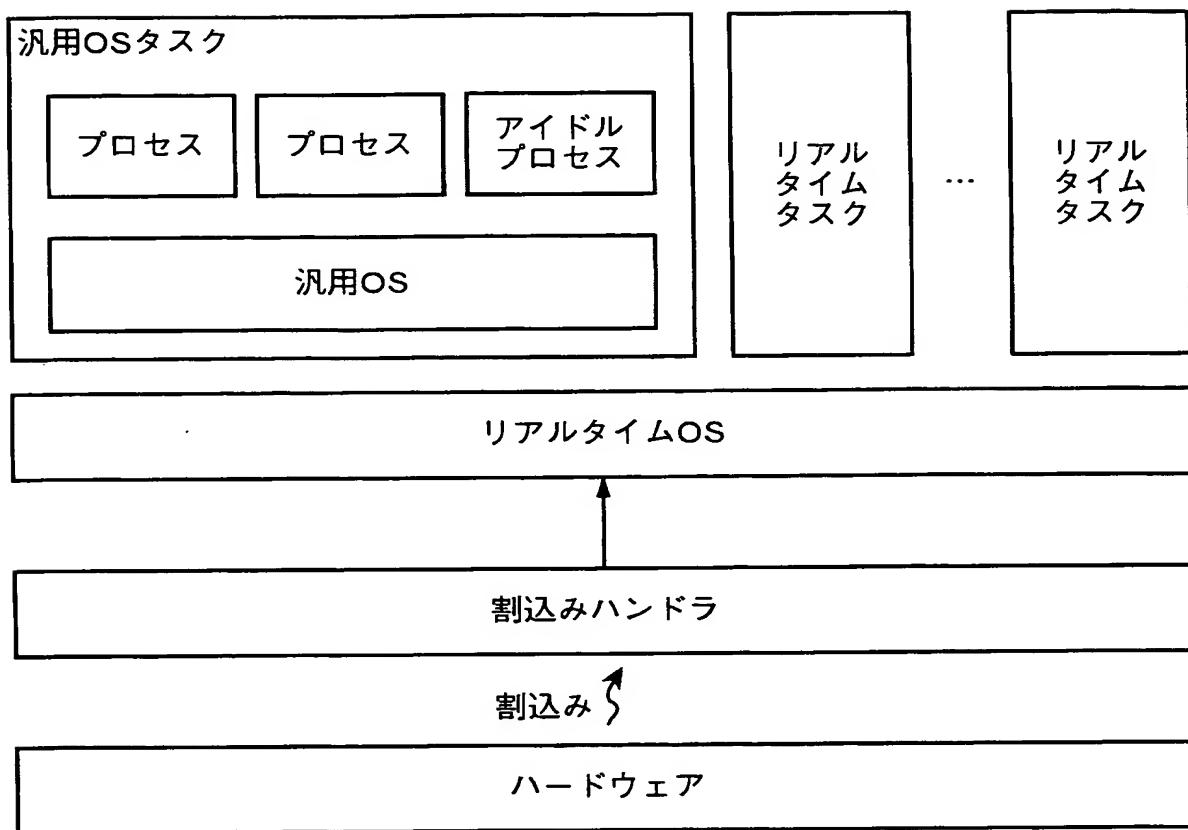
【図 10】

OS間負荷処理均衡化に関する汎用OSタスクの状態遷移を示す図

汎用OSタスクの状態遷移と、OS間負荷処理均衡化のためのリタルマ/OS間連操作								
イベント/条件	状態							
	S1	S2	S3	S4	S5	操作	操作	操作
遷移先	初期	処理必要(本優先度)	処理必要(均衡化優先度)	遷移先	操作	遷移先	操作	遷移先
E1 タスク生成	S2	・実優先度=本優先度 ・持続タイマ=持続タイマ/7起動	-	-	-	-	-	-
E2 タイマ込み	-	-	S2	持続タイマ-1	-	-	S4	II 均衡化実行タイマ-1
E3 持続タイマ=0	-	-	S3	・実優先度=均衡化優先度 ・均衡化実行タイマ=均衡化実行時間	-	-	-	-
E4 タスクブチク	-	-	S4	持続タイマ停止	S4	均衡化実行タイマ作動	S2	・実優先度=本優先度 ・持続タイマアカ起動
E5 均衡化実行タイマ=0	-	-	-	-	-	-	S2	・実優先度=本優先度 ・持続タイマアカ
E6 資源待ち	-	-	-	-	-	-	S5	・実優先度=本優先度 ・均衡化実行タイマ停止
E7 資源待ち解除	-	-	-	-	-	-	-	S2
E8 均衡化要求	-	-	S3	・実優先度=均衡化優先度 ・均衡化実行タイマ=均衡化実行時間	-	-	-	-

【図11】

ハイブリッドOSを説明するための説明図



【書類名】要約書

【要約】

【課題】オペレーティングシステムを一つのタスクとしてコンピュータに実行させるタスク管理プログラムでオペレーティングシステムとの間で負荷処理の均衡化を図り、システム全体のスループットを向上すること。

【解決手段】タイマ割込みハンドラ130がPCBリスト310および割込み処理要求フラグ320を用いて汎用OSタスク400の処理の必要性を判定し、汎用OSタスク400の処理が必要な場合には、汎用OSタスク400の優先度を上げることとする。

【選択図】

図1

特願2003-301573

出願人履歴情報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社